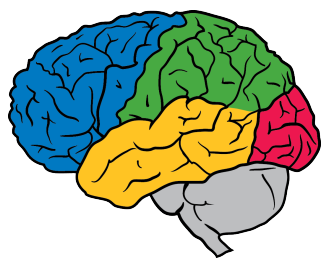
Size-free generalization bounds for convolutional neural networks

Hanie Sedghi

Google Brain

joint work with Phil Long



Generalization bounds for DNNs

- Substantial progress in theoretical analysis of the generalization of deep learning models [Zhang et al., 2016, Dziugaite and Roy, 2017, Bartlett et al., 2017, Neyshabur et al., 2017, 2018, Arora et al., 2018, Neyshabur et al., 2019].
- [Bartlett, 1998]: Even if there are many parameters, the set of models computable using weights with small magnitude is limited enough to provide leverage for induction [Bartlett et al., 2017, Neyshabur et al., 2018].
- There is a tendency for these algorithms to produce small weights (implicit bias in deep learning). [Gunasekar et al., 2017, 2018,, Ma et al., 2018].

Distance to initialization

- Generalization bounds in terms of the distance from the initial setting of the weights instead of the size of the weights [Bartlett et al., 2017, Neyshabur et al., 2019].
- Small initial weights may promote vanishing gradients; Instead, choose initial weights that maintain a strong but non-exploding signal as computation flows through the network [LeCun et al., 2012, Glorot and Bengio, 2010, Saxe et al., 2013, He et al., 2015].
- For a large network initialized in this way, a variety of well-behaved functions can be found through training by traveling a short distance in parameter space [Du et al., 2019,, Allen-Zhu et al., 2019].
- The distance from initialization may be expected to be significantly smaller than the magnitude of the weights. Furthermore, there is theoretical reason to expect that, as the number of parameters increases, the distance from initialization decreases.

Generalization bounds for CNNs

- Convolutional layers are used in all competitive deep neural network architectures applied to image processing tasks.
- The most influential generalization analyses in terms of distance from initialization have so far concentrated on networks with fully connected layers.
- A convolutional layer has an alternative representation as a fully connected layer, earlier analyses apply in the case of convolutional networks
- But, intuitively, the weight-tying employed in the convolutional layer constrains the set of functions computed by the layer.
- This additional restriction should be expected to aid generalization.

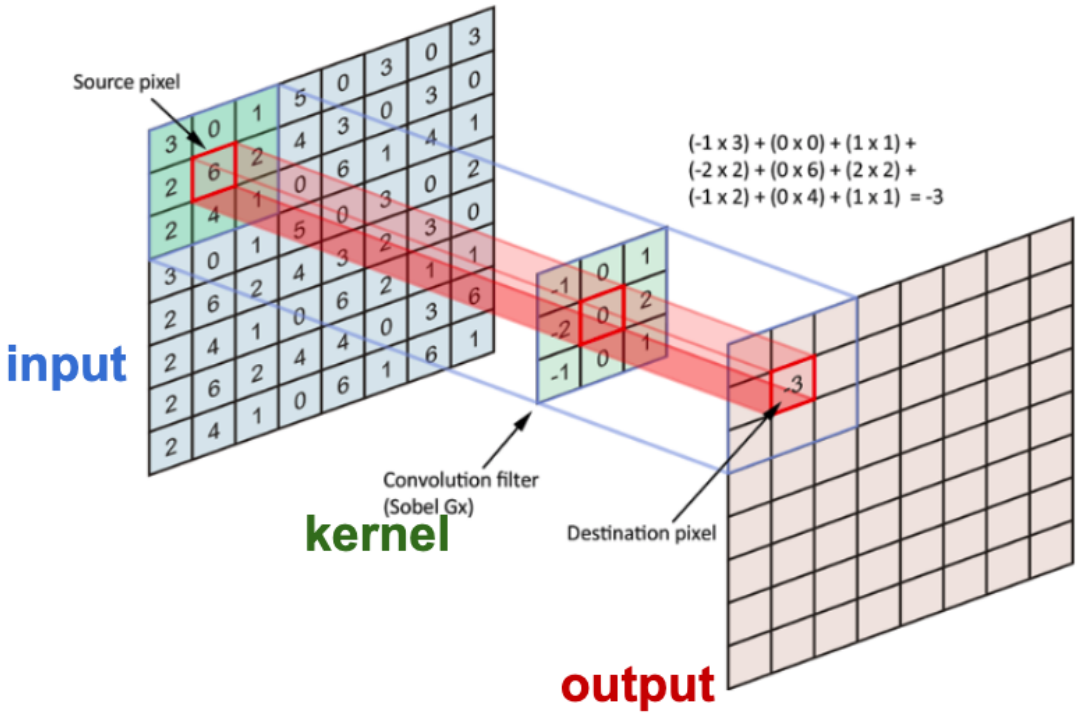
Summary of Our Results

- Our bounds are in terms of
  - the training loss,
  - the number of parameters
  - the Lipschitz constant of the loss
  - distance from the weights to the initial weights.
- They are independent of
  - the number of pixels in the input,
  - the height of hidden feature maps;
  - the width of hidden feature maps;

The first supervised learning bounds for deep convolutional networks with this property.

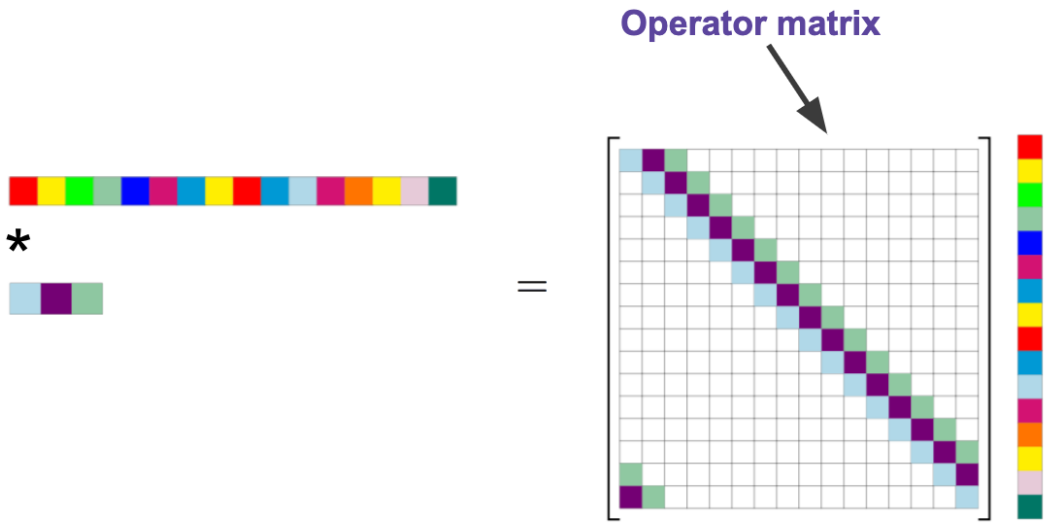
Recap: Convolution

$$\forall ij,\ Y_{ij} = \sum_{p\in[n]} \sum_{q\in[n]} X_{i+p,j+q} K_{p,q}$$



Convolution: A linear Operator

1D Convolution



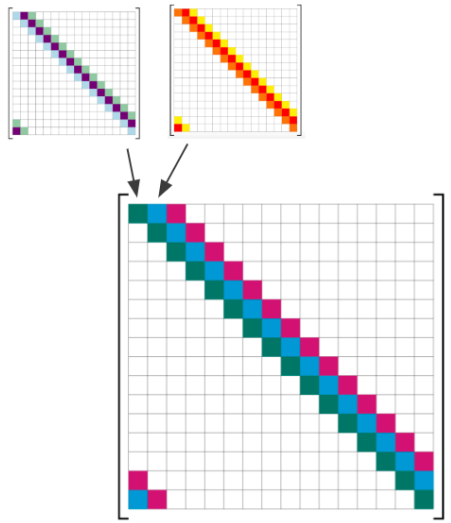
$$\forall i,\ Y_i = \sum_{p \in [n]} X_{i+p} K_p$$

Operator matrix $A = \mathrm{op}(K)$

Convolution: A linear Operator

2D Single channel Convolution

$$Y_{ij} = \sum_{p \in [n]} \sum_{q \in [n]} X_{i+p,j+q} K_{p,q}$$



$$A = \begin{bmatrix} \operatorname{circ}(K_{0,:}) & \operatorname{circ}(K_{1,:}) & \dots & \operatorname{circ}(K_{n-1,:}) \\ \operatorname{circ}(K_{n-1,:}) & \operatorname{circ}(K_{0,:}) & \dots & \operatorname{circ}(K_{n-2,:}) \\ \vdots & \vdots & \vdots & \vdots \\ \operatorname{circ}(K_{1,:}) & \operatorname{circ}(K_{2,:}) & \dots & \operatorname{circ}(K_{0,:}) \end{bmatrix}$$

$$= \operatorname{op}(K)$$

Spectral Analysis of Convolution [Sedghi et al., 2019]

- Analyzed 2D multi-channel convolution.
- Proposed simple, efficient algorithm to find the spectrum.
- Proposed upper bounds for spectral norm of 2D multi-channel convolution.
- Proposed an algorithm for projecting a convolutional layer onto an operator-norm ball.
- Can be extended to 3D multi-channel convolution.

H. Sedghi, V. Gupta and P. Long, *The Singular Values of Convolutional Layers*, ICLR 2019

Bounds for a basic setting

- Zero-padding, pooling, the activations are 1-Lipschitz and nonexpansive (e.g, ReLU, tanh)
- Input $x \in \mathbb{R}^{d \times d \times c}$, $\| \operatorname{vec}(x) \| \leq 1$.
- Number of channels c, Kernels $K^{(i)} \in \mathbb{R}^{k \times k \times c \times c}, \quad \forall i \in [L]$
- $W = Lk^2c^2$ total no. of parameters

Bounds for a basic setting

- Zero-padding, pooling, the activations are 1-Lipschitz and nonexpansive (e.g, ReLU, tanh)
- Input $x \in \mathbb{R}^{d \times d \times c}$, $\| \operatorname{vec}(x) \| \leq 1$.
- Number of channels c, Kernels $K^{(i)} \in \mathbb{R}^{k \times k \times c \times c}$, $\forall i \in [L]$
- $W = Lk^2c^2$ total no. of parameters
- Loss function $\ell : \mathbb{R} \times \mathbb{R} \to [0, 1]$, $\ell(\cdot, y)$ is λ-Lipschitz for all y.

Bounds for a basic setting

- Zero-padding, pooling, the activations are 1-Lipschitz and nonexpansive (e.g, ReLU, tanh)
- Input $x \in \mathbb{R}^{d \times d \times c}$, $\| \operatorname{vec}(x) \| \leq 1$.
- Number of channels c, Kernels $K^{(i)} \in \mathbb{R}^{k \times k \times c \times c}, \quad \forall i \in [L]$
- $W = Lk^2c^2$ total no. of parameters
- Loss function $\ell : \mathbb{R} \times \mathbb{R} \to [0, 1]$, $\ell(\cdot, y)$ is λ-Lipschitz for all y.
- $|| \operatorname{op}(K_0^{(i)})||_2 = 1, \quad \forall i \in [L]$
- $||K - K_0||_\sigma \stackrel{\text{def}}{=} \sum_{i=1}^{L} || \operatorname{op}(K^{(i)}) - \operatorname{op}(K_0^{(i)})||_2$.
- $F_\beta = \{f_K : ||K - K_0||_\sigma \leq \beta\}$.

Bounds for a basic setting

Theorem (Basic bounds)

For any $\eta > 0$, there is a $C > 0$ such that for any $\beta \geq 5$, $\lambda \geq 1$, $\delta > 0$, for any joint probability distribution P over $\mathbb{R}^{d \times d \times c} \times \mathbb{R}$, if a training set S of n examples is drawn independently at random from P, then, with probability at least $1 - \delta$, for all $f \in F_\beta$,

$$\mathbb{E}_{z \sim P}[\ell_f(z)] \leq (1 + \eta)\mathbb{E}_S[\ell_f(z)] + \frac{C(W(\beta + \log(\lambda n)) + \log(1/\delta))}{n}$$

and

$$\mathbb{E}_{z \sim P}[\ell_f(z)] \leq \mathbb{E}_S[\ell_f(z)] + C\sqrt{\frac{W(\beta + \log(\lambda)) + \log(1/\delta)}{n}}.$$

Tools and Proof Outline

Definition

For $d \in \mathbb{N}$, a norm over $\mathbb{R}^d$ is *full* if its unit ball has positive volume.

Definition

For $d \in N$, a set G of functions with a common domain Z, we say that G is (B, d)-*Lipschitz parameterized* if there is a full norm $||\cdot||$ on $\mathbb{R}^d$ and a mapping ϕ from the unit ball w.r.t. $||\cdot||$ in $\mathbb{R}^d$ to G such that, for all θ and θ' such that $||\theta|| \leq 1$ and $||\theta'|| \leq 1$, and all $z \in Z$, $|(\phi(\theta))(z) - (\phi(\theta'))(z)| \leq B||\theta - \theta'||$.

Tools and Proof Outline

Lemma [Vapnik and Chervonenkis, 1971, Vapnik, 1982, Pollard, 1984, Giné and Guillou, 2001]
A set G of functions $g : Z \to [0, M]$ is (B, d)-Lipschitz parameterized. Then, for any $\eta > 0$, there is a C such that, for all large enough $n \in \mathbb{N}$, for any $\delta > 0$, with probability at least $1 - \delta$, for all $g \in G$,

$$\mathbb{E}_{z\sim P}[g(z)] \leq (1+\eta)\mathbb{E}_S[g] + \frac{CM(d\log(Bn) + \log(1/\delta))}{n}$$

and

$$\mathbb{E}_{z\sim P}[g(z)] \leq \mathbb{E}_S[g] + CM\sqrt{\frac{d\log B + \log(1/\delta)}{n}}.$$

Tools and Proof Outline

Lemma [Vapnik and Chervonenkis, 1971, Vapnik, 1982, Pollard, 1984, Giné and Guillou, 2001]
A set G of functions $g : Z \to [0, M]$ is (B, d)-Lipschitz parameterized. Then, for any $\eta > 0$, there is a C such that, for all large enough $n \in \mathbb{N}$, for any $\delta > 0$, with probability at least $1 - \delta$, for all $g \in G$,

$$\mathbb{E}_{z\sim P}[g(z)] \leq (1+\eta)\mathbb{E}_S[g] + \frac{CM(d\log(Bn) + \log(1/\delta))}{n}$$

and

$$\mathbb{E}_{z\sim P}[g(z)] \leq \mathbb{E}_S[g] + CM\sqrt{\frac{d\log B + \log(1/\delta)}{n}}.$$

We show ℓ_{F_β} is $\left(\beta\lambda e^{\beta}, W\right)$-Lipschitz parameterized.

A closer look

- The old [Vapnik and Chervonenkis, 1971, Vapnik, 1982, Pollard, 1984]

$$\mathbb{E}_{z\sim P}[g(z)] \leq (1+\eta)\mathbb{E}_S[g] + \frac{CM(d\log(Bn)+\log(1/\delta))}{n}$$

A closer look

- The old [Vapnik and Chervonenkis, 1971, Vapnik, 1982, Pollard, 1984]

$$\mathbb{E}_{z\sim P}[g(z)] \leq (1+\eta)\mathbb{E}_S[g] + \frac{CM(d\log(Bn)+\log(1/\delta))}{n}$$

- The newer! [Giné and Guillou, 2001]

$$\mathbb{E}_{z\sim P}[g(z)] \leq \mathbb{E}_S[g] + CM\sqrt{\frac{d\log B+\log(1/\delta)}{n}}.$$

A closer look

- The old [Vapnik and Chervonenkis, 1971, Vapnik, 1982, Pollard, 1984]

$$\mathbb{E}_{z\sim P}[g(z)] \leq (1+\eta)\mathbb{E}_S[g] + \frac{CM(d\log(Bn)+\log(1/\delta))}{n}$$

- The newer! [Giné and Guillou, 2001]

$$\mathbb{E}_{z\sim P}[g(z)] \leq \mathbb{E}_S[g] + CM\sqrt{\frac{d\log B+\log(1/\delta)}{n}}.$$

- The secret?

A closer look

- The old [Vapnik and Chervonenkis, 1971, Vapnik, 1982, Pollard, 1984]

$$\mathbb{E}_{z\sim P}[g(z)] \leq (1+\eta)\mathbb{E}_S[g] + \frac{CM(d\log(Bn)+\log(1/\delta))}{n}$$

- The newer! [Giné and Guillou, 2001]

$$\mathbb{E}_{z\sim P}[g(z)] \leq \mathbb{E}_S[g] + CM\sqrt{\frac{d\log B+\log(1/\delta)}{n}}.$$

- The secret?
- When using covering bounds of the form $\left(\frac{B}{\epsilon}\right)^d$, they paid particular attention to the dependence of the resulting generalization bound on B.

Lipschitz Parametrization

Parameter change in one layer

- ℓ is λ-Lipschitz w.r.t. its first argument,
  $|\ell(f_K(x), y) - \ell(f_{\tilde{K}}(x), y)| \leq \lambda |f_K(x) - f_{\tilde{K}}(x)|$,
- Bound $|f_K(x) - f_{\tilde{K}}(x)|$.

Lipschitz Parametrization

Parameter change in one layer

- ℓ is λ-Lipschitz w.r.t. its first argument,
  $|\ell(f_K(x), y) - \ell(f_{\tilde{K}}(x), y)| \leq \lambda |f_K(x) - f_{\tilde{K}}(x)|,$
- Bound $|f_K(x) - f_{\tilde{K}}(x)|$.
  $f_K = g_{\text{down}} \circ f_{\text{op}(K^{(j)})} \circ g_{\text{up}}$.
  $u = g_{\text{up}}(x)$, $\|u\| \leq \prod_{i<j} \left\| \text{op}(K^{(i)}) \right\|_2$.

Lipschitz Parametrization

Parameter change in one layer

- ℓ is λ-Lipschitz w.r.t. its first argument,
  $|\ell(f_K(x), y) - \ell(f_{\tilde{K}}(x), y)| \leq \lambda |f_K(x) - f_{\tilde{K}}(x)|$,
- Bound $|f_K(x) - f_{\tilde{K}}(x)|$.
  $f_K = g_{\mathsf{down}} \circ f_{\mathrm{op}(K^{(j)})} \circ g_{\mathsf{up}}$.
  $u = g_{\mathsf{up}}(x)$, $\|u\| \leq \prod_{i<j} \left\|\mathrm{op}(K^{(i)})\right\|_2$.

$$
\begin{aligned}
|f_K(x) - f_{\tilde{K}}(x)| &= |g_{\mathsf{down}}(\mathrm{op}(K^{(j)})u) - g_{\mathsf{down}}(\mathrm{op}(\tilde{K}^{(j)})u)| \\
&\leq \left(\prod_{i \neq j} \left\|\mathrm{op}(K^{(i)})\right\|_2\right) \left\|\mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)})\right\|_2 \\
&\leq \left(\prod_{i \neq j} (1 + \beta_i)\right) \left\|\mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)})\right\|_2
\end{aligned}
$$

Lipschitz Parametrization

- Parameter change in one layer

$$
\begin{aligned}
|\ell(f_K(x), y) - \ell(f_{\tilde{K}}(x), y)| &\leq \lambda \left(\prod_{i \neq j} (1 + \beta_i) \right) \left\| \mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)}) \right\|_2 \\
&\leq \lambda (1 + \beta/L)^L \left\| \mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)}) \right\|_2 \\
&\leq \lambda e^{\beta} \left\| \mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)}) \right\|_2 .
\end{aligned}
$$

Lipschitz Parametrization

- Parameter change in one layer

$$
\begin{aligned}
|\ell(f_K(x), y) - \ell(f_{\tilde{K}}(x), y)| &\leq \lambda \left(\prod_{i \neq j} (1 + \beta_i) \right) \left\| \mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)}) \right\|_2 \\
&\leq \lambda (1 + \beta/L)^L \left\| \mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)}) \right\|_2 \\
&\leq \lambda e^{\beta} \left\| \mathrm{op}(K^{(j)}) - \mathrm{op}(\tilde{K}^{(j)}) \right\|_2 .
\end{aligned}
$$

- Change in all layers:
  - one layer at a time
  - triangle inequality

$$
|\ell(f_K(x), y) - \ell(f_{\tilde{K}}(x), y)| \leq \lambda e^{\beta} \left\| K - \tilde{K} \right\|_{\sigma} .
$$

ℓ_{F_β} is $\left(\beta \lambda e^{\beta}, W\right)$-Lipschitz parameterized.

Comparison to [Bartlett et al., 2017]

- Parametrize convolution as fully connected
- h.p bound on $\mathbb{E}_{z\sim P}[\ell_f(z)] - \mathbb{E}_S[\ell_f(z)]$.

Comparison to [Bartlett et al., 2017]

- Parametrize convolution as fully connected
- h.p bound on $\mathbb{E}_{z\sim P}[\ell_f(z)] - \mathbb{E}_S[\ell_f(z)]$.
- Simplify: Initialization computes Identity, $K = K_0 + \epsilon\mathbb{1}$
  - Our bound

$$\frac{c^{3/2}kL + ck\sqrt{\log(\lambda)} + \sqrt{\log(1/\delta)}}{\sqrt{n}}.$$

  - [Bartlett et al., 2017] bound

$$\frac{(c+1)^L\sqrt{c}d(d/k)^{3/2}L^{3/2}\lambda\log(dcL) + \sqrt{\log(1/\delta)}}{\sqrt{n}}$$

  - In this scenario, the new bound is independent of d, and grows more slowly with λ, c and L.

A more general setting

- Zero-padding, pooling activations are nonexpansive (e.g., ReLU , tanh)
- L_c convolutional layers, L_f fully connected layers.
- $x \in \mathbb{R}^{d \times d \times c}$, $\| \operatorname{vec}(x) \| \leq \chi$, and $y \in \mathbb{R}^{d \times d \times c}$,
- $\ell(\cdot, y)$ is λ-Lipschitz for all y and that $\ell(\hat{y}, y) \in [0, M]$ for all $\hat{y}$ and y.
- $|| \operatorname{op}(K_0^{(i)})||_2 \leq 1 + \nu$, and for all fully connected layers i, $||V_0^{(i)}||_2 \leq 1 + \nu$.

A more general setting

- Notation
  - $V^{(i)}$: weights for the ith fully connected layer.
  - $\Theta = (K^{(1)}, ..., K^{(L_c)}, V^{(1)}, ..., V^{(L_f)})$ all parameters
  - $L = L_c + L_f$.
  - $||\Theta - \tilde{\Theta}||_N = \left(\sum_{i=1}^{L_c} ||\operatorname{op}(K^{(i)}) - \operatorname{op}(\tilde{K}^{(i)})||_2\right) + \sum_{i=1}^{L_f} ||V^{(i)} - \tilde{V}^{(i)}||_2$.
  - $\mathcal{F}_{\beta,\nu} = \{f_\Theta : ||\Theta - \tilde{\Theta}||_N \leq \beta\}$.

General bound

Theorem (General Bound)

For any $\eta > 0$, there is a constant C such that the following holds. For any $\beta, \nu, \chi > 0$ such that $\chi\lambda\beta e^{\beta} \geq 5$, for any $\delta > 0$, for any joint probability distribution P over $\mathbb{R}^{d \times d \times c} \times \mathbb{R}^m$ such that, with probability 1, $(x, y) \sim P$ satisfies $||\operatorname{vec}(x)||_2 \leq \chi$, if a training set S of n examples is drawn independently at random from P, then, with probability at least $1 - \delta$, for all $f \in \mathcal{F}_{\beta,\nu}$,

$$\mathbb{E}_{z\sim P}[\ell_f(z)] \leq (1+\eta)\mathbb{E}_S[\ell_f(z)] + \frac{CM\left(W\left(\beta + \nu L + \log(\chi\lambda\beta n)\right) + \log(1/\delta\right.}{n}$$

and,

$$\mathbb{E}_{z\sim P}[\ell_f(z)] \leq \mathbb{E}_S[\ell_f(z)] + CM\sqrt{\frac{W\left(\beta + \nu L + \log(\chi\lambda\beta)\right) + \log(1/\delta)}{n}}.$$

Corollary

- $||K - K_0||_\sigma \leq ||\operatorname{vec}(K) - \operatorname{vec}(K_0)||_1$ [Sedghi et al., 2019].
- Same bounds can be reached if the definition of F_β is replaced with the analogous definition using $||\operatorname{vec}(\Theta) - \operatorname{vec}(\tilde{\Theta}_0)||_1$.

Corollary

- $||K - K_0||_\sigma \leq ||\operatorname{vec}(K) - \operatorname{vec}(K_0)||_1$ [Sedghi et al., 2019].
- Same bounds can be reached if the definition of F_β is replaced with the analogous definition using $||\operatorname{vec}(\Theta) - \operatorname{vec}(\tilde{\Theta}_0)||_1$.
- Bound holds uniformly for models at a distance β from initialization – can be modified using standard techniques to get nonuniform bound in terms of distance.

Another Comparison: Fully connected case

- $D = cd^2$, each hidden layer has D components, D classes.
- For all i, $V_0^{(i)} = I$ and $V^{(i)} = I + H/\sqrt{D}$, H is a Hadamard matrix.
- Each layer V, $||V||_2 = 2$, $||V - V_0||_2 = 1$, $||V - V_0||_{2,1} = D$.

Another Comparison: Fully connected case

- $D = cd^2$, each hidden layer has D components, D classes.
- For all i, $V_0^{(i)} = I$ and $V^{(i)} = I + H/\sqrt{D}$, H is a Hadamard matrix.
- Each layer V, $||V||_2 = 2$, $||V - V_0||_2 = 1$, $||V - V_0||_{2,1} = D$.

- Our bound

$$\frac{DL + D\sqrt{L\log(\lambda)} + \sqrt{\log(1/\delta)}}{\sqrt{n}}$$

- [Bartlett et al., 2017] bound

$$\frac{\lambda 2^L L^{3/2} D \ln(DL) + \sqrt{\log(1/\delta)}}{\sqrt{n}}$$

Experiments: CIFAR10

- Setting
  - VGG style 10-layer all-convolutional model
  - CIFAR10 dataset
  - dropout, exponential learning rate schedule.
  - repeatedly trained for different values of batch size and initial learning rate.

Generalization gap $\stackrel{\text{def}}{=}$ Difference between train and test error

Generalization gap

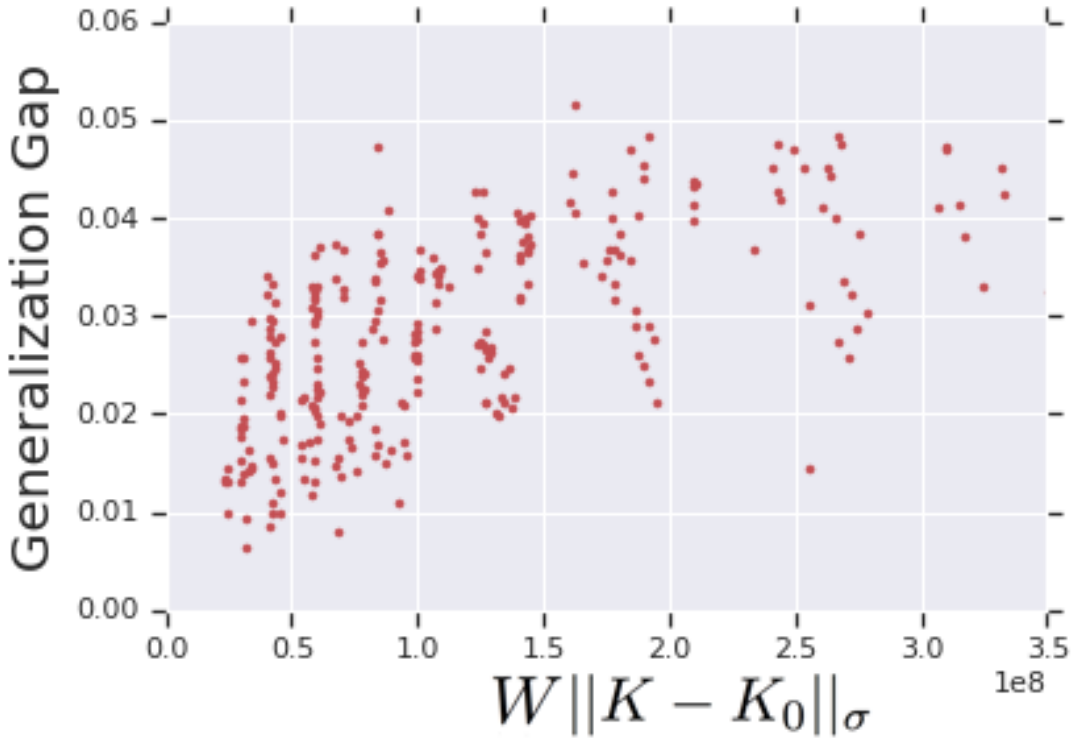
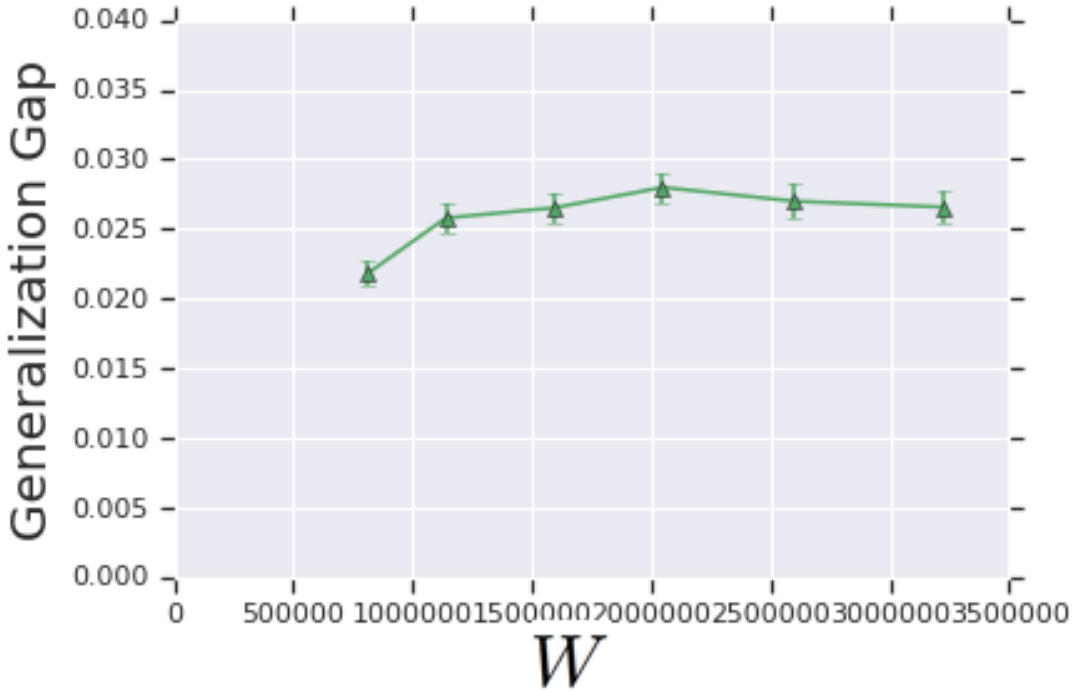


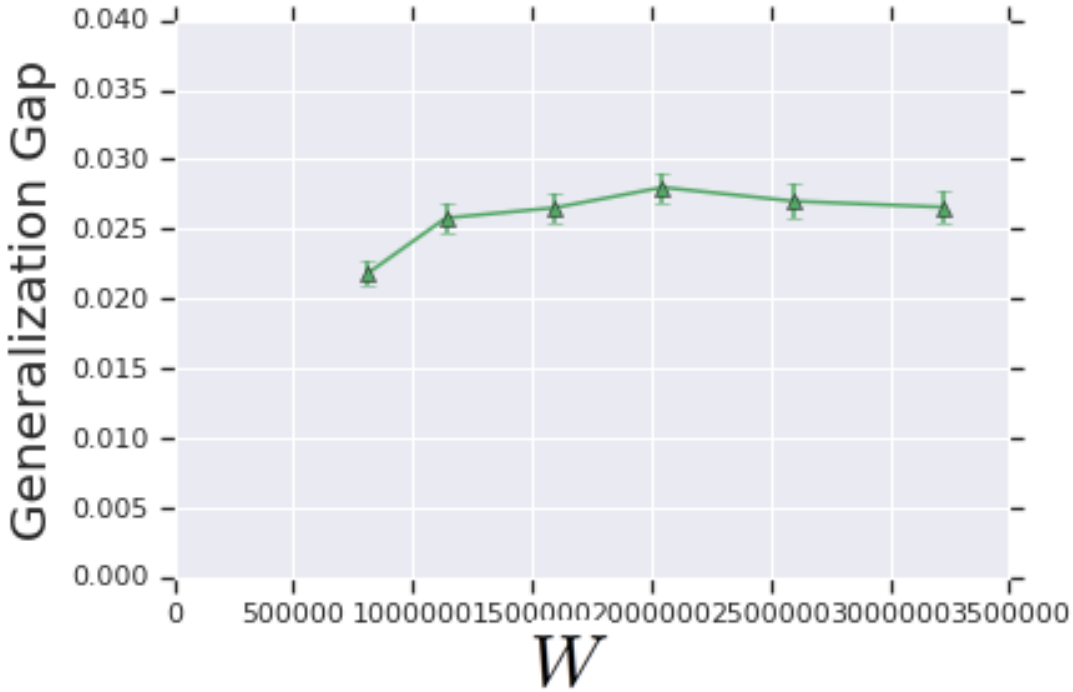
Figure: Generalization gaps for a 10-layer all-conv model on CIFAR10 dataset.

Generalization gap as a function of W



We increase no. of parameters by making the network wider.

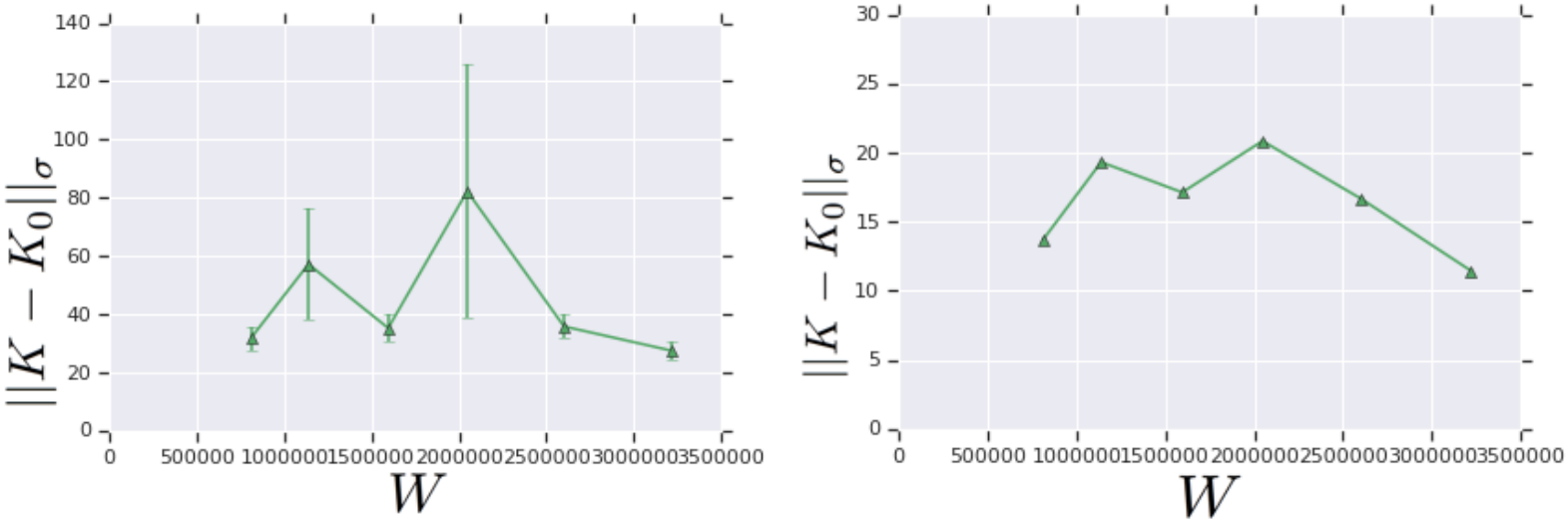
Generalization gap as a function of W



We increase no. of parameters by making the network wider.

As the network becomes more over-parametrized,the generalization gap remains almost flat.

Distance to initialization as a function of W

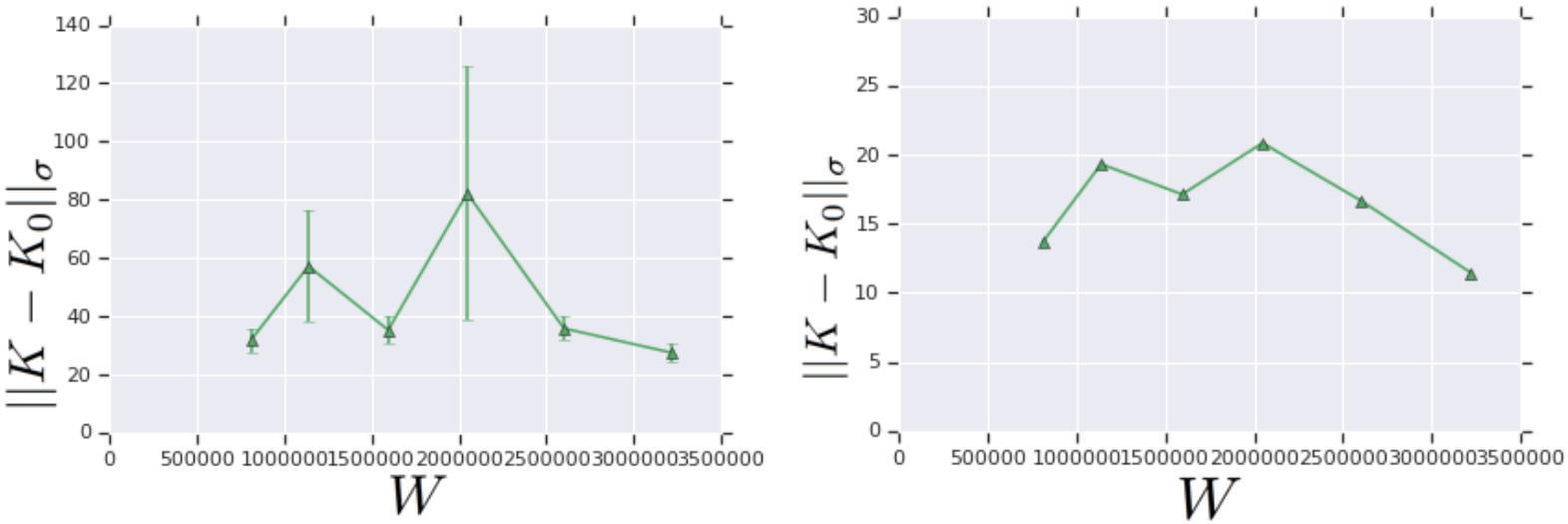


(a) mean and error bar

(b) median

Figure: $||K - K_0||_\sigma$ as a function of W.

Distance to initialization as a function of W



(a) mean and error bar

(b) median

Figure: $||K - K_0||_\sigma$ as a function of W.

Increasing W leads to a decrease in value of $||K - K_0||_\sigma$.

Experiments: Role of input size

- Downsampled CIFAR-10 images from 32×32 to 16×16

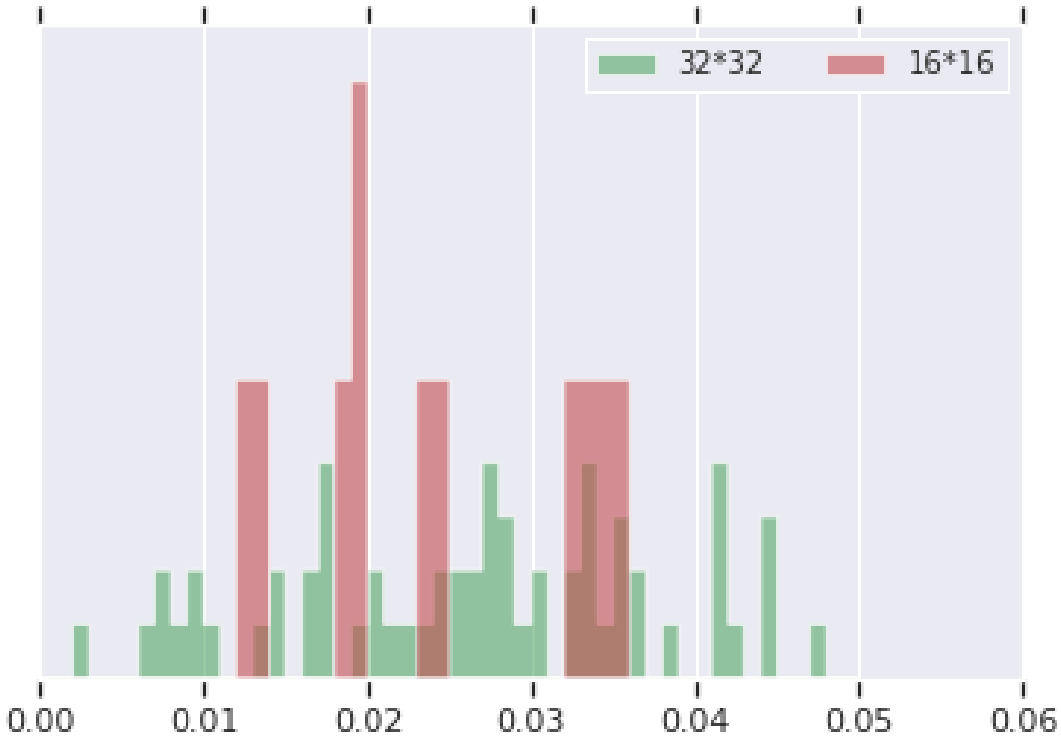


Figure: Generalization gaps of 10-layer conv models

Experiments: Role of input size

- Downsampled CIFAR-10 images from 32×32 to 16×16

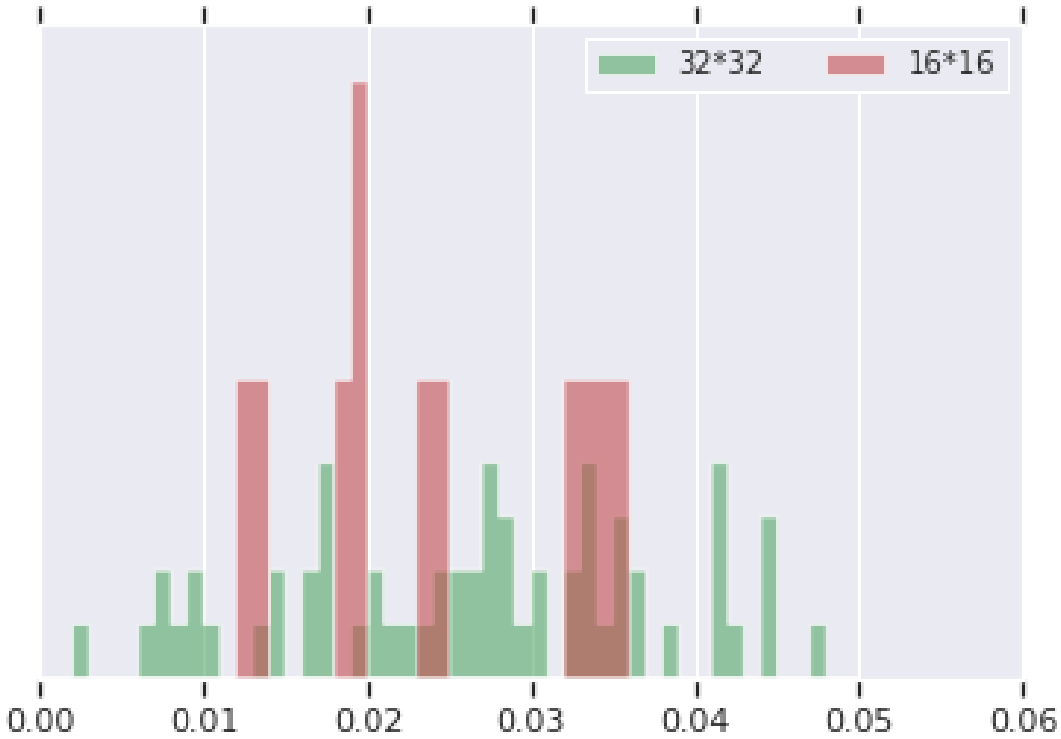


Figure: Generalization gaps of 10-layer conv models

Generalization gap does not depend on the input size.
Our bounds capture this.

Conclusion and Future Work

- First size-free generalization bounds for deep CNN models.
- Our analysis applies to practical architectures.
  - The activation functions and pooling operators can have larger Lipschitz constants.
- Many variants of our bounds are possible.
  - We have chosen to present relatively simple and interpretable bounds.
  - bounds in terms of Lipschitz constants of subnetworks which are bounded above by e^{β}.
- Future work: use the insights for better training.

Conclusion and Future Work

- First size-free generalization bounds for deep CNN models.
- Our analysis applies to practical architectures.
  - The activation functions and pooling operators can have larger Lipschitz constants.
- Many variants of our bounds are possible.
  - We have chosen to present relatively simple and interpretable bounds.
  - bounds in terms of Lipschitz constants of subnetworks which are bounded above by e^{β}.
- Future work: use the insights for better training.

Pre-print available on arXiv.

Conclusion and Future Work

- First size-free generalization bounds for deep CNN models.
- Our analysis applies to practical architectures.
  - The activation functions and pooling operators can have larger Lipschitz constants.
- Many variants of our bounds are possible.
  - We have chosen to present relatively simple and interpretable bounds.
  - bounds in terms of Lipschitz constants of subnetworks which are bounded above by e^{β}.
- Future work: use the insights for better training.

Pre-print available on arXiv.

Thank You!

References I

[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *ICML*, 2019.

[2] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.

[3] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

[4] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[5] Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *ICML*, 2019.

[6] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *ICLR*, 2019.

[7] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *UAI*, 2017.

[8] Evarist Giné and Armelle Guillou. On consistency of kernel density estimators for randomly censored data: rates holding uniformly over adaptive intervals. In *Annales de l'IHP Probabilités et statistiques*, volume 37, pages 503–522, 2001.

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[10] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6151–6159, 2017.

[11] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018.

[12] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pages 9461–9471, 2018.

References II

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[14] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[15] Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In *ICML*, pages 3351–3360, 2018.

[16] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[17] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *ICLR*, 2018.

[18] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *ICLR*, 2019.

[19] D. Pollard. *Convergence of Stochastic Processes*. Springer Verlag, Berlin, 1984.

[20] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[21] Hanie Sedghi, Vineet Gupta, and Philip M Long. The singular values of convolutional layers. *ICLR*, 2019.

[22] V. N. Vapnik. *Estimation of Dependencies based on Empirical Data*. Springer Verlag, 1982.

[23] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[24] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.